IJE available online at www.ije-pyc.org
INDONESIAN JOURNAL OF ENERGY

Applying Artificial Neural Network and XGBoost to Improve Data Analytics in Oil and Gas Industry

Ricky Simanjuntak [1,]*, Dedy Irawan [1]

[1] *Petroleum Engineering Study Program, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Jalan Ganesha No. 10, Bandung 40132, Indonesia*

Received 9 July 2020; Accepted 19 August 2020
Available online 26 February 2021

Abstract. The application of machine learning and artificial intelligence is popular nowadays to improve data analytics in the oil and gas industry. A huge amount of data can be processed to gain insights about the subsurface conditions, even reducing time for manual review or interpretation. There are three cases to be discussed in this study that starts from porosity estimation of thin core image using Otsu's thresholding, estimation of oil production rate from sucker-rod pumping wells and sonic travel-time log generation. Two supervised learning algorithms are applied, XGBoost and Keras. These algorithms will capture all possible correlations between the input and output data. From data normalization, exploratory data analysis and model building, the workflow is built on Google Colab. The original dataset is split into training and testing. Tuning hyperparameters such as the number of hidden layers, neurons, activation function, optimizers and learning rates are captured to reduce the complexity of the model. The model is evaluated by error values and the coefficient of determination to estimate the model skill on unseen data.

Keywords: model, machine, regression

1. Introduction

1.1 Background

The profitability of oil wells will decrease along with the production declines, but the costs for workover, maintenance and development will become much higher (Achmad, 2017). The implementation of digital field technology has become a continuous effort in major oil and gas companies around the world integrating the latest advances in AI-based technology to improve well performance, production monitoring and business results. Digital solutions affect the analysis, simulation, scenarios and decision. A digital oil well is equipped with a downhole monitoring system and sensors generating data every second to monitor the real-time condition of the wells. By implementing technology related to digitalization also artificial intelligence, field data can bring more intuition to support the decision workflows and optimize oil production. From the simplest application, to determine the porosity of a thin core image. A simple core image will be interpreted as a two-dimensional matrix of a pixel, and the algorithm just needs to calculate the total number of white and black pixels. The total flow rate of wells with the artificial lift sucker-rod pump is measured, but the contributions of the individual wells are unknown. It is critical to know the single contributions to account material balance, well monitoring, reservoir management and future economic analysis. Oil-flow measurement (Equation 1) can be performed using orifice meters. A range of metering equipment and techniques are available to provide high accuracy flow rate measurements under a wide range of conditions. For steady-state conditions, the most commonly used is the orifice flow meter. A multiphase flow meter is expensive to install and operate, but it provides more valuable information by determining the flow rates of different phase components in a flowing stream through one device (Campos et al., 2014). For a larger number of wells, the flow tests are infrequent. Monitoring of production is not

*Corresponding author
E-mail address: rickysm@students.itb.ac.id

usually carried out in the real-time and a generalized form of Gilbert correlation can be used to allocate production to the wells (Khan et al., 2019). The last is the sonic log generation. Sonic travel time logs are quite important for subsurface characterization around the wellbore (Bukar et al., 2019). However, due to operational costs issues or constraints, these logs are not always acquired or only run in certain wells. Gamma-ray, density, neutron and resistivity are known as conventional logs that are run in most of the wells. If sonic logs are not acquired in a well, they can be synthesized based on existing data or neighbor wells and paired with their subsurface properties from conventional logs.

$$q_o = \frac{P_{wh} D^{A_3}}{A_1 R^{A_2}} \tag{1}$$

Where,

q_o	=	Oil flow rate, stb/day
P_{wh}	=	Wellhead pressure, psia
D	=	Choke size, in
R	=	Gas-liquid-ratio, Mscf/stb
A_1, A_2, A_3	are constants	

1.2 Supervised Machine Learning Algorithm

Supervised machine learning (Figure 1) is one of the types of machine learning algorithms that trains both the input and output data (Caruana & Niculescu-Mizil, 2006). The targets are known; for example, in the case of sucker-rod pumping wells, the oil flowrates from the separator test are acquired. Sonic travel time logs from a limited number of wells will be paired with conventional logs and are trained through the learning process. Keras and XGBoost methods are used in this study. Keras is a high-level Application Programming Interface (API) with the model and layers as the main structure. The most time-consuming process is to determine the optimum architecture of the model to minimize the error (Li et al., 1995). XGBoost or Extreme Gradient Boosting is a widely used machine learning method in data science and machine learning competitions because it performs well in most data sets (Chen & Guestrin, 2016).

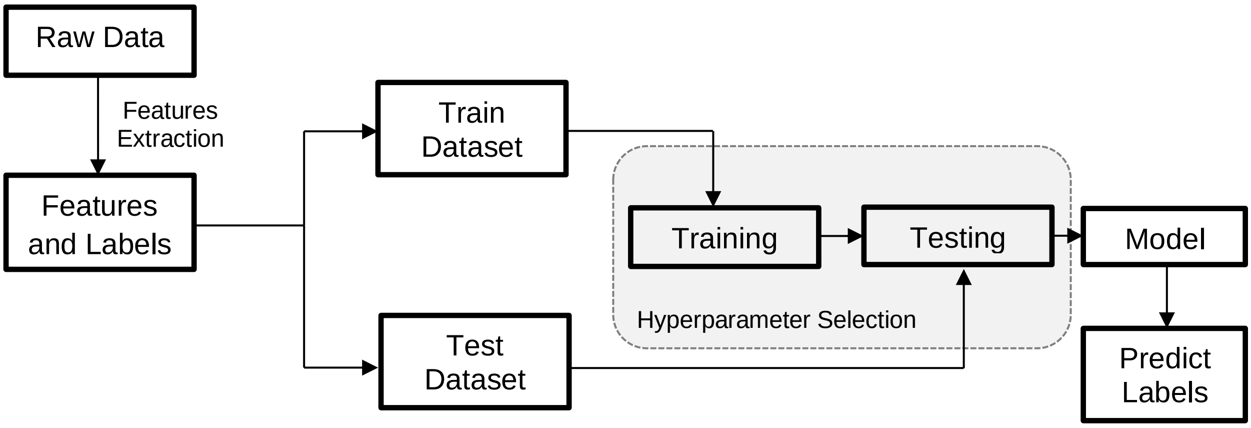


Figure 1. Supervised machine learning workflow.

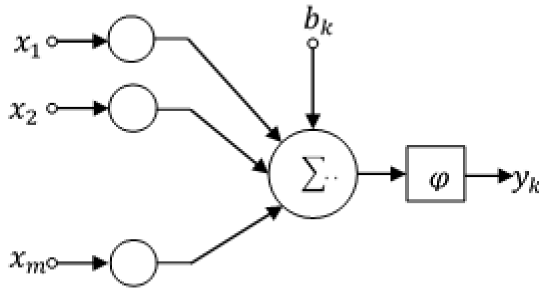


Figure 2. Artificial neural network model.

The idea of an artificial neural network (ANN) model (Figure 2) is that it begins with random weights, then will be adjusted through iteration and backpropagation to reduce the error (Equation 2).

$$y_k = \mathbf{f}\left(\sum_{j=1}^{m} w_{kj}x_j + b_k\right) \quad (2)$$

f	=	activation functions
x_j	=	input
w_{kj}	=	weights of neutron
b_k	=	bias
y_k	=	output

2. Methodology

2.1 Data Collection

The data is obtained from Field "X" with an artificial lift sucker-rod pump equipped with a dynamometer for a downhole card survey. The first step is to collect all input variables, and they are plunger size, stroke length, pump displacement, pumping speed, pump fillage, minimum polished rod load (MPRL), peak polished rod load (PPRL), card area and barrel oil per day (BOPD). Pumping speed is measured as strokes per minute, the number of strokes the polished rod completes in one minute. For conventional pumping units, the stroke length is available with a maximum of 240 inches, but long-stroke pumping units can have up to 366 inches. When idealized conditions are assumed to prevail, the plunger is completely filled with fluids. Incomplete filling (less than 100%) usually occurs when pump capacity is higher than the inflow rate of the well and can be happened due to gas interference (Takacs, 2015). The statistics descriptions are shown in Table 1. For the case of porosity estimation, the input data is a thin core image, as shown in Figure 3. When using a wireline log to determine the formation characteristics, often one of the curves was missing caused by tool failure or was not included during the measurement. To address this problem, several studies using machine learning have been performed to generate synthetic well logs. Input data for the sonic log generation is in Table 2. The data is obtained with a total sample of 30,143 before normalization. There are seven input parameters, caliper log, neutron log, gamma-ray, deep resistivity, medium resistivity, photo-electric factor, density log and two outputs, compressional travel-time (DTC) and shear travel-time (DTS).

Table 1. Input parameters of sucker-rod pumping wells.

Variable	Min.	Max.	Mean
Plunger size, in	1.75	3.75	2.96
Pumping speed, spm	5.77	11.42	8.88
Stroke length, in	68	100	87.26
Pump fillage, %	11.2	100	71.19
MPRL, lbs.	656	8,496	2,038.8
PPRL, lbs.	2,096	10,752	5,664.9
Pump Disp, B/D	183	1,841.8	593.16
Card area, in.lbs.	168	8,094.7	1,239.6
Barrel oil per day	24.1	1,660.8	412.75

Table 2. Input parameters of sonic log generation.

Variable	Min.	Max.	Mean
Caliper log, in	5.93	21.06	8.42
Neutron log	0.014	0.998	0.235
Gamma-ray, API	1.038	199.676	43.041
Deep Res, ohm/m	0.129	124.249	2.549
Med Res, ohm/m	0.185	914.191	2.7293
Photo E, Barn/e	0.002	28.106	3.918
Density, gr/cm^3	0.68	3.25	2.415
DTC, $\mu s/ft$	49.97	155.98	87.69
DTS, $\mu s/ft$	80.58	487.44	181.33

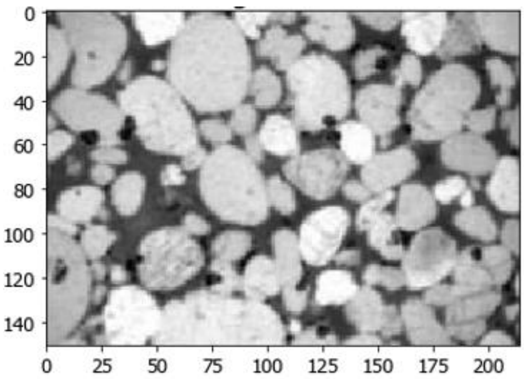


Figure 3. Core image sample for porosity estimation. From the photomicrograph of a quartz arenite (sandstone) St. Peter Formation (Ordovician) near Dixon, Illinois (Rygel, 2010).

2.2 Model Building

The data is rarely homogenous and there are two major problems, outliers and missing values. These will reduce the performance of the model. Feature scaling is an important part of data pre-processing before applying machine learning algorithms. By scaling inputs variables, gradient descent would not take a very long time to converge. Deciding on the architecture of neural networks is very time-consuming and complicated. Since there is no clear theory to determine the number of nodes in each hidden layer or the number of layers, the common practice uses trial and error. Several researchers try to determine the optimum architecture by the equation shown in Table 3 where N_h is the number of neurons in a hidden layer, $n = N_i =$ input neurons, N_o is output neurons, and N is total samples.

Table 3. Various approaches for the architecture.

No.	Various methods	Year	Number of hidden neurons
1	Li et al. method	1995	$N_h = \left(\sqrt{1+8n} - 1\right)/2$
2	Tamura and Tateishi method	1997	$N_h = n - 1$
3	Zhang et al. method	2003	$N_h = 2^n/(n+1)$
4	Xu and Chen method	2008	$N_h = C_f(N/n\,log\,N)^{0.5}$
5	Shibata and Ikeda method	2009	$N_h = \sqrt{N_i N_o}$
6	Hunter et al. method	2012	$N_h = 2^n - 1$

For hyperparameter tuning of the XGBoost model, Grid Search is very popular. Consider a grid space of hyperparameter stored in a dictionary. For each hyperparameter pair, this tool will conduct cross-validation on the training set with a certain number of training samples per iteration and choose the parameter that leads to the minimum error (Table 4).

Table 4. Grid search algorithm (Doma & Pirouz, 2020).

```
Algorithm: Parameter tuning using Grid Search
Input: $Hyperparameters\ Dictionary$
Output: $best\ parameters\ key: value$
import dataset
for $i\ \in\ values\ of\ each\ key \in$
$parameter\ Dictionary$ do
  for $n\ fold\ cross\ validation$ do
      get maximum accuracy value
      if $i > LastParams$ then
          $i > LastParams\ =\ i$
      End
  End
End
```

Dropout is a powerful technique that can be applied to an artificial neural network (Figure 4) to prevent overfitting. Overfitting is the overtraining of limited training data, which results in learning complicated models and performs poorly on new data or the test data set. Dropout works by randomly dropping out units of the hidden layer (Figure 5). Dropout rate 0.2 is used, which means one in five neurons in the hidden layer will be randomly excluded from each update cycle. The parameters for the artificial neural network model are shown in Table 5.

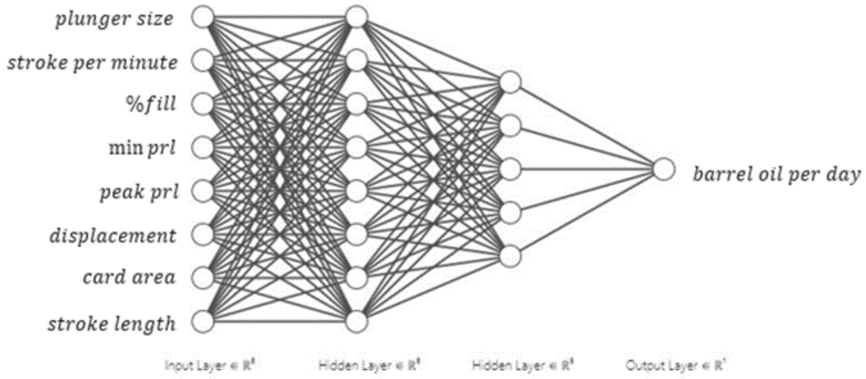


Figure 4. Artificial neural network architecture.

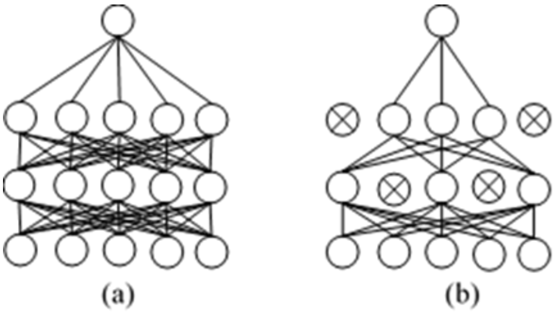
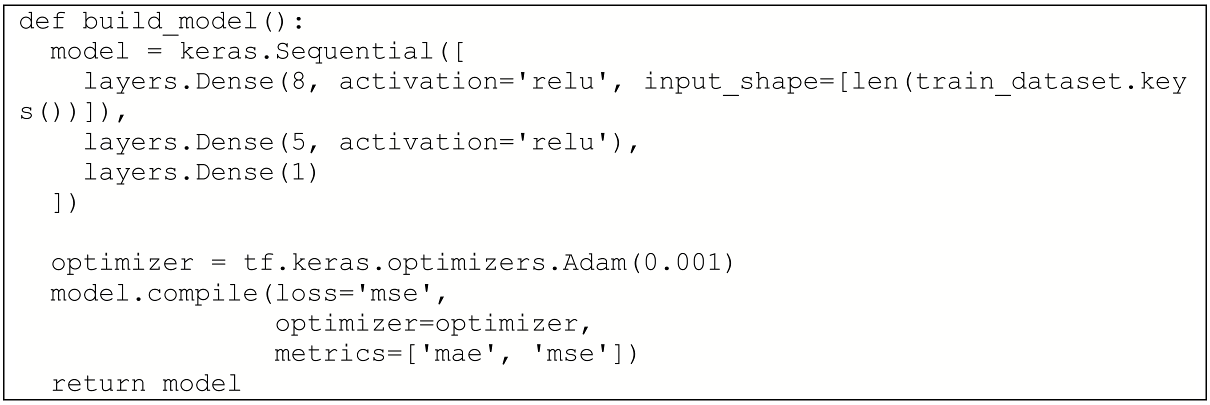


Figure 5. Without dropout rate (a) and after dropout (b).

Table 5. Design parameters for ANN Model.

Parameter	Value
Output neuron	1
No. of hidden layer	2
Input neurons	8
No. of epochs	100
Weight initializer	Glorot normal
Activation function	ReLU, Linear
Learning rate	0.1
Batch size	32
Optimizer	Adam
Dropout rate	0



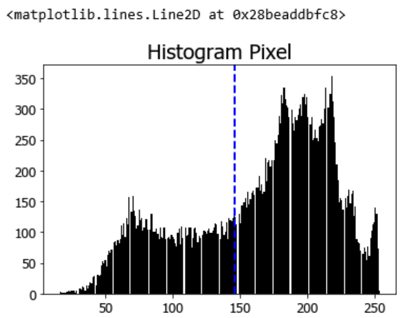
```
def build_model():
  model = keras.Sequential([
    layers.Dense(8, activation='relu', input_shape=[len(train_dataset.key
s())]),
    layers.Dense(5, activation='relu'),
    layers.Dense(1)
  ])

  optimizer = tf.keras.optimizers.Adam(0.001)
  model.compile(loss='mse',
                optimizer=optimizer,
                metrics=['mae', 'mse'])
  return model
```

Figure 6. Keras model after hyperparameters tuning.

3. Results and Discussion

3.1 Case A: Porosity Estimation

A sample core image is applied to Gaussian Blur to reduce the noise within the image. Python will interpret the core as a two-dimensional matrix of a pixel from 0 to 255. The distribution of pixels after Gaussian Blur is shown in Figure 7. The blue dash line is the chosen threshold by Otsu's thresholding, and only black and white color will be displayed (Figure 8). The porosity will be calculated from the number of white and black pixels (Figure 9). The porosity is 0.3178.



```
<matplotlib.lines.Line2D at 0x28beaddbfc8>
```

Figure 7. Gaussian blur.

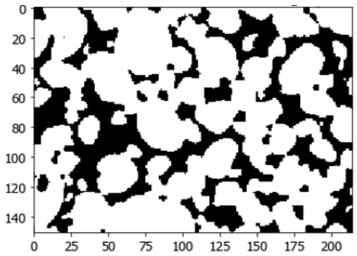


Figure 8. Core image after Otsu's Thresholding.

```
blur = cv2.GaussianBlur(img,(5,5),0)
ret, th = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
n_white_pix = np.sum(th == 255)
n_black_pix = np.sum(th == 0)
porosity = n_black_pix/(n_white_pix+n_black_pix)
```

Figure 9. Porosity estimation by calculating the number of white and black pixels.

3.2 Case B: Estimating Oil Production Rates

After training the model with a train size of 80%, the model is tested to see the performance. The coefficient of determination evaluates the model skill, denoted as R^2. Results for the training period, the R^2 is 0.9994, and R^2 is 0.9752 for the testing period (Figure 10). Residual plots display the residual values on the y-axis and the actual barrel oil per day on the x-axis (Figure 11). The points are very close to the fitted line and the residuals center on zero, indicated a good fit of the regression model.

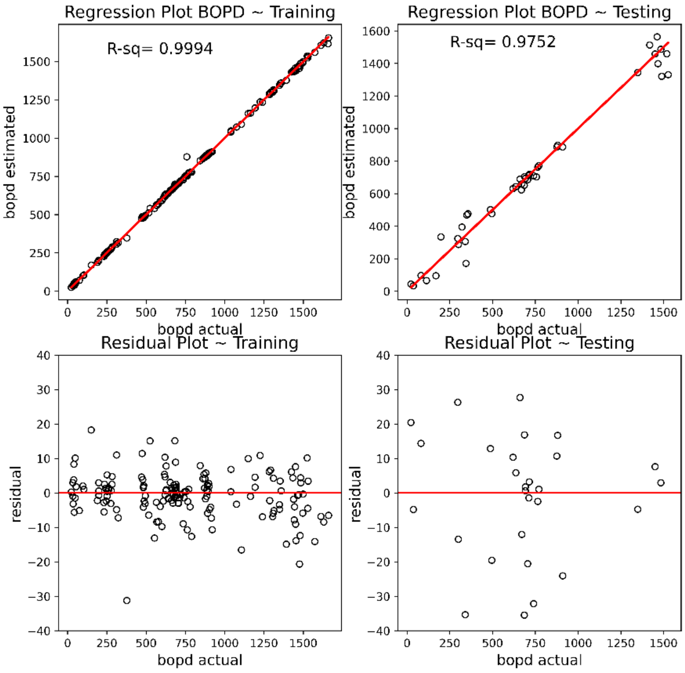


Figure 10. Regression and residual plot.

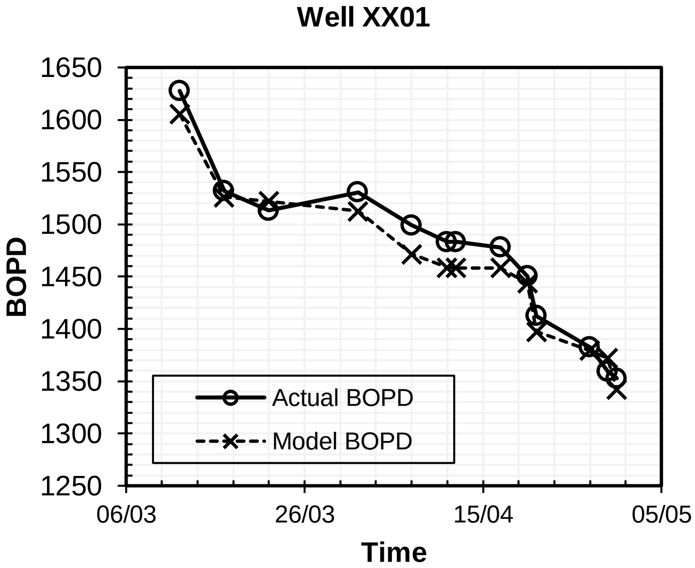


Figure 11. Production rates of sucker-rod pumping wells.

3.3 Case C: Pseudosonic Log Generation

The sonic log is generated using XGBoost, a gradient boosting decision trees algorithm for the efficiency of computing time and memory resources (Figure 12). The best hyperparameters are chosen using Grid Search, a tool for hyperparameters tuning with cross-validation on Google Colab to faster the running time. Since there are two targets (DTC and DTS), as shown in Figure 13, the results will be summarized in Table 6 and Table 7. Root-mean-square error (RMSE) is defined as the equation below (Equation 3).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left[y^{\mathrm{pred}(i)} - y^{(i)}\right]^2}{n}} \tag{3}$$

```
model= xgboost.XGBRegressor(n_estimators=100, learning_rate=0.1,
                            objective='reg:squarederror',
                            gamma=0.6, subsample=0.8,
                            colsample_bytree=0.8,max_depth=10,
                            min_child_weight=3,
                            random_state=42)
model.fit(train$x,train$y)
```

Figure 12. XGBoost model in Google Colab.

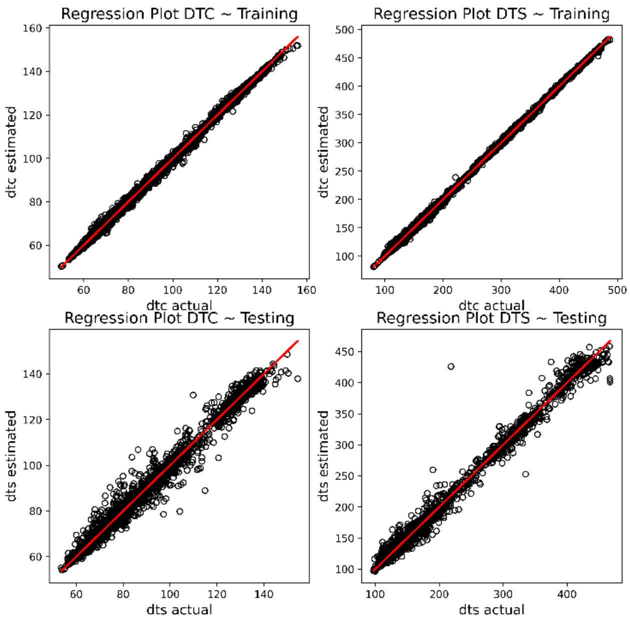


Figure 13. Regression plot for DTC and DTS.

Table 6. Coefficient of determination.

Target	Training	Testing
DTC	0.99904495711231	0.98753707866924
DTS	0.99954584882725	0.99102019201806

Table 7. Root mean square error.

Target	Training	Testing
DTC	0.728889373741041	2.63487665174514
DTS	1.820038472606744	8.17570127875725

4. Conclusion

This study shows some of the applications of supervised machine learning algorithms, from the simplest one, porosity estimation and to the regression cases, oil flowrates and sonic logs. The artificial neural network can accurately estimate the oil production rates by using the online dynamometer data. Extreme gradient boosting algorithm, XGBoost, works very well in generating the compressional travel-time and shear travel-time logs. It is emphasized that the model is trained on a certain range of data, and estimating the target by using some instances outside the range will not give a valid estimation. The hyperparameters of the two supervised machine learning algorithms can be improved. The model should be trained because these hyperparameters might have changed after inputting other variables and a significant amount of data.

References

Achmad, Z. (2017). Value creation of digital oilfield technology, study case Duri Field well monitoring. *SPE/IATMI Asia Pacific Oil & Gas Conference and Exhibition*. https://doi.org/10.2118/186903-ms

Bukar, I., Adamu, M. B., & Hassan, U. (2019). A machine learning approach to shear sonic log prediction. *SPE Nigeria Annual International Conference and Exhibition*. https://doi.org/10.2118/198764-ms

Campos, S., Baliño, J., Slobodcicov, I., Filho, D., & Paz, E. (2014). Orifice plate meter field performance: Formulation and validation in multiphase flow conditions. *Experimental Thermal and Fluid Science*, *58*, 93-104. https://doi.org/10.1016/j.expthermflusci.2014.06.018

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine learning - ICML '06*. https://doi.org/10.1145/1143844.1143865

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Doma, V., & Pirouz, M. (2020). A comparative analysis of machine learning methods for emotion recognition using EEG and peripheral physiological signals. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00289-7

Khan, M. R., Alnuaim, S., Tariq, Z., & Abdulraheem, A. (2019). Machine learning application for oil rate prediction in artificial gas lift wells. *SPE Middle East Oil and Gas Show and Conference*. https://doi.org/10.2118/194713-ms

Li, J., Chow, T., & Ying-Lin Yu. (1995). The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network. *Proceedings of ICNN'95 - International Conference on Neural Networks*. https://doi.org/10.1109/icnn.1995.487330

Rygel, M. C. (2010). *Photomicrograph of a quartz arenite (sandstone) from the St. Peter Formation (Ordovician) near Dixon, Illinois*.
https://commons.wikimedia.org/wiki/File:Quartz-arenite.jpg

Takacs, G. (2015). *Sucker-rod pumping handbook: Production engineering fundamentals and long-stroke rod pumping*. Gulf Professional Publishing.